(4)

IBM

International Business Machines Corporation

Department B18/Building 931
Box 390, Poughkeepsie, N. Y. 12602
914/463-1234
Direct Dialing No.: (914) 485-9915

1970 March 2

ESC/111

Mr. L. M. Burstein
FAA RD-630
Washington, D. C. 20509

Dear Lloyd:

I must commence with a disclaimer. What follows in this letter are my own personal judgments. They are not official positions or recommendations of the IBM Corporation, and I trust you not to consider or represent them to be such.

The proposed American National Standard Keyboards I shall refer to as the pANS Keyboards. In view of the negative ballots at the ANSI X4 Committee, the pANS is now under review by the ANSI Subcommittee X4-A9. In such review situations, the potential for changing the pANS exists. In order to assess the "changeability" of the pANS, it is worthwhile to review the history.

I shall use the terms "bit-paired" and "typewriter-paired", although you do not applaud the appropriateness of those terms in your write-up. I use them, because appropriate or not, we both understand what is meant when I use them. (More on "appropriateness" later, by the way).

The fundamental dichotomy in the two pANS keyboard-layouts arose from the location of the "(" and ")" in ASCII. In ASCII, they were paired with the numerics "8" and "9". On typewriters, both manual and electric, they were (and are) paired with the numerics "9" and "0". Given that the parentheses were paired with "9" and "0" on typewriters, why did they not become so paired in the code?

Back in 1962, when ANSI (then ASA) Subcommittee X3.2 was developing ASCII, an international code standard was also being developed, under ISO/TC97/SC2. USA manufacturers, who market their products internationally, were dedicated to making the ISO Code and the USA Code the same, in order to be able to manufacture one line of products world-wide.

With respect to the location of the parentheses in the ISO Code, the USA ran up against the hard fact that many European typewriters paired the parentheses with the "8" and "9". The Europeans were dedicated to them being so paired in the code.

The USA, therefore, was faced with a difficult choice. Since the Europeans could not be dissuaded from pairing the parentheses with the "8" and "9" in the code, the only way to achieve international agreement was to accept this pairing. I assure you, Lloyd, having been deeply involved in those standards committee meetings, both ISO/TC97/SC2 and ANSI X3.2, it was not a USA decision easily arrived at. The pros and cons were discussed intensively and extensively with strong and able protagonists on both sides. Regardless of the ultimate merits of the decision, the USA did decide to accept the European requirement, in order to achieve international agreement on the code. Thus began the time of troubles for USA keyboard standardization.

CBI 81/20/50

The "(" and ")" would be paired with the "8" and "9" in the code, and in consequence, would be bit-paired on a bit-paired keyboard. Why should they not become so paired on typewriters? Two reasons spoke compellingly against.

Reason 1. Touch-typing is not a uniform skill with typists. All of them touch-type the alphabetics, the numerics, and certain special graphics such as the period and comma. The good touch-typist touch-types all special graphics, but the average (i.e., vast majority) touch-typist touch-types only those specials which occur frequently in her (or his) normal office correspondence. The infrequently used specials are not, in fact, touch-typed, but are typed by hunt-and-peck. Typewriter manufacturers have studied and researched typists' habits and patterns exhaustively, and they will tell you that certain specials are frequently used, and in consequence, are nearly always touch-typed even by average typists. These specials are left parenthesis, right parenthesis, dash (or hyphen), quotation marks. Next in frequency of usage come asterisk, slash, apostrophe, semi-colon, etc. Beyond that, it depends on the office correspondence applications. For example, in an office where there is heavy typing of name and address, the ampersand will be touch-typed.

What I am saying here, Lloyd, is that the typewriter manufacturers have proved conclusively to themselves that the left and right parentheses must never be moved from their traditional nine and zero keys on typewriters.

Reason 2. Two graphics in the code have a unique characteristic. The numeric zero, and the underscore, do not bit-pair with any other graphic. This has been a thorn in the side of the bit-pairing enthusiasts. If the bit-pairing rule is to be maintained with complete rigor, then the only solution is to have the numeric zero alone on a key, and the underscore alone on a key. And so it was done in the pANS. However, this required 48 keys. If cost consideration dictate no more than 47 keys, then jettison the bit-pairing rule on one key only: i.e., put the numeric zero and the underscore on the same key. And so it was done in the pANS, as an option.

But the typewriter manufacturers have a different consideration. With the advent of the electric typewriter, certain keys became repeat keys when held down. Very quickly, typewriter manufacturers realized that the dash and the underscore were very desirable as repeat graphics, and they put them on one key. As you may be aware, the underscore had been, and indeed still is, on the upper case numeric six key on manual typewriters. The placement of the underscore on a repeat key for electric typewriters was the basic factor that changed the electric typewriter keyboard layout from the manual typewriter keyboard layout.

In any event, the typewriter manufacturers had another reason for retaining the right parenthesis on the numeric zero key. With the right parenthesis on the numeric zero key, the underscore could not go there, and could remain paired with the dash on a repeat key.

In summary then, the bit-paired enthusiasts were and are locked in on the numeric eight and nine keys, while the typewriter manufacturers were and are locked in on the left and right parentheses on the numeric nine and zero keys.

I call this the basic dichotomy between the bit-paired and typewriter-paired keyboards.

Let me review the agreements and disagreements between the two pANS keyboard layouts.

There are 94 graphics plus Space in the code. I will categorize the graphics as:

a) **Hard core**, which means their location is identical on both layouts.

b) **Dichotomy**, which means their locations are different on the two layouts.

c) **Low usage**, which means that their attribute of low usage is more significant than whether or not they occupy the same locations on the two layouts.

Hard Core Set

A - Z	26
a - z	26
Ø - 9	10
#$%- .,;?! < > /\ [] { }	17
Total	79

Dichotomy Set

( ) * + = ^ ' " _ : & @	12

Low Usage Set

` ! ~	3

The Hard Core Set is shown on Figure 1.

It is unfortunate that the ~ ^ graphics are not on the same key in both layouts, since there seems to be no reason why they should not be.

At this point, it is a temptation to see if greater agreement could be reached. Accepting that the bit paired protagonists and the typewriter-paired protagonists cannot be persuaded away from their convictions on left parenthesis, right parenthesis, and underscore, so that a fundamental dichotomy does and will exist, is it possible to arrive at compromise keyboards which differ only in the location of these graphics?

We start with Figure 2, the incomplete bit-paired layout, and complete it according to typewriter pairing, giving Figure 4, Compromise 1. Then we start with Figure 3, the incomplete typewriter-paired layout, and complete it according to bit-pairing, giving Figure 6, Compromise 2. For comparison purposes, Figure 5 shows the Typewriter-Paired Layout, and Figure 7 shows the Bit-Paired Layout.

We now compare Figures 4 and 5, and compare Figures 6 and 7.

	Figures 4 & 5		Figures 6 & 7	
Hard Core	A - Z a - z Ø - 9 #$%- .,;?! <> /\ [] { }	26 26 10 17	A - Z a - z Ø - 9 #$%- .,;?! <> /\ [] { }	26 26 10 17
Now Agree	@+:" \ ~ ^ = '	9	@+:" \ ~ ^ * ¦	9
Still Disagree	&()_*¦	6	&()_='	6
		94		94

We see that we have, for both Compromise 1 and 2, increased the agreement from 79 to 88 graphics.

Given that we cannot persuade the protagonists to give up on the basic dichotomy on left parenthesis, right parenthesis, underscore, so that we would still have to live with <u>two</u> standard keyboard layouts, would the bit-pairing enthusiasts accept Compromise 1 to replace the Bit Paired Layout of Figure 7, or alternatively, would the typewriter-pairing enthusiasts accept Compromise 2 to replace the Typewriter-Paired Layout of Figure 5?

I judge the answer to both questions would be "No!!!".

For the bit paired enthusiasts, the fundamental criterion is that <u>all</u> keys must be bit-paired. While they might accept one <u>key</u> to be non-bit-paired, (the zero and underscore key) where cost considerations are important with respect to the nmber of keys, they would never (I believe) accept as many non-bit-paired keys as are shown on Compromise 1.

For the typewriter paired enthusiasts, the fundamental criterion is that as many as possible of the graphics be typewriter-paired. So they would never (I believe) accept Compromise 2, which has many non-typewriter paired keys.

I have gone through the exercise above to demonstrate my convictions that a <u>single</u> compromise keyboard will never be acceptable to <u>both</u> groups. The pragmatic aspect of the situation is that <u>both</u> groups have accepted the philosophy that <u>both</u> kinds of keyboard layouts be in the standard.

You may be interested in the kind of analysis I perform on proposed keyboard layouts. It is a purely pragmatic analysis, and is based on the consideration that many (although not all, in fact, far from a majority) of IBM keyboard devices are based on the IBM Selectric. So one thing I look for is the set of graphics that fall within the 44-key area (keys numbered 1 through 44 on your keyboard layout charts). Secondly, I determine which of the 63 plus Space graphics from the center four columns of ASCII are, and which are not, in the 44-key area.

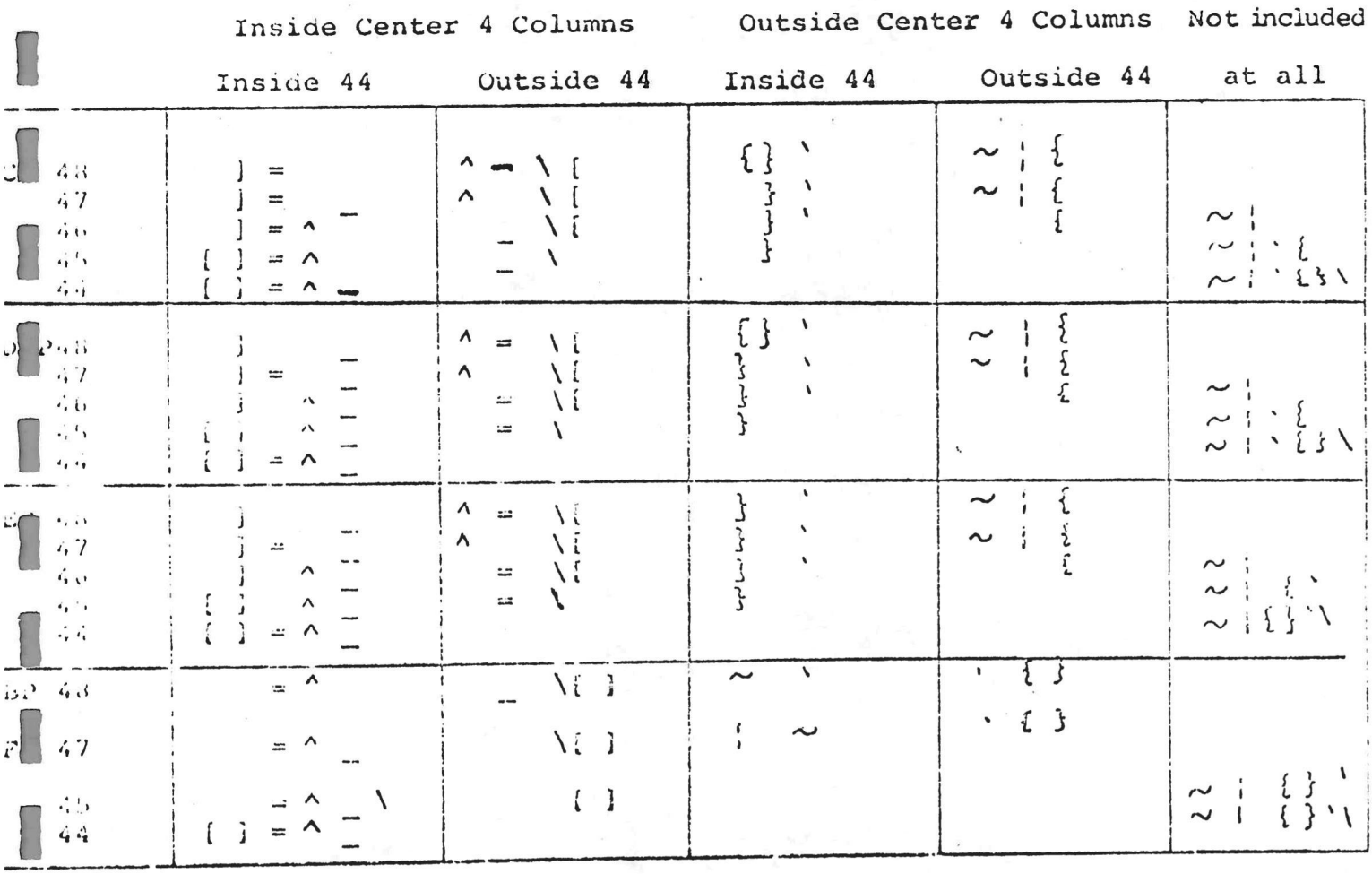
Simple arithmetic demonstrates that 26 of these graphics will be the lower case alphabet, leave 62 other graphic positions. Clearly then, one of the 63 graphics in the center four columns of ASCII cannot be in the 44-key area.

In the following chart I have focused only on a particular set of special graphics; namely,

[ ] = ^ _ \ ` { | } ~

I have focused on these because in CP, D4P, EP, BP, and FP configurations for 48, 47, 46, 45, 44 keys, they are sometimes in the 44-key area and sometimes not.

If I arbitrarily decide that the single graphic in the center four columns of ASCII that should not be in the 44-key area is "\", I then draw some conclusions on your layouts. I would favor <u>none</u> of the 48, 47, 46, 45 key configurations, but would favor all of the 44-key configurations.



This leads me to ask a question. All of your 44-key configurations have one characteristic in common. They all include in the 44-key area all of the graphics of the center four columns except the "\". Why not, then, take this as your basic configuration, or more correctly as your basic configurations, one for bit-pairing, one for typewriter-pairing, and build up the 45, 46, 47, 48 key configurations, keeping the 44-key areas immutable? Surely then, you develop a set of keyboards which are consistent in the locations of the graphics of the center four columns of ASCII (standing aside from the basic dichotomy of bit-pairing versus typewriter pairing) and in which only the "\" and the specials from columns 6 and 7 move around. My intuition tells me that this would lead to a more viable set of keyboards than those in which graphics like __/ | | move around.

I now address semantics, bit-paired, typewriter-paired, code-paired, early-paired, etc. I submit that your designation "code-paired" suffers from the same defect you point out for "typewriter-paired". The so-called typewriter-paired keyboard is intended also for keyboard devices which will produce coded output. The designation "early-paired" does not suffer from this defect. However, I feel it is not sufficiently suggestive. Indeed "typewriter-paired" and "bit-paired" are both suggestive of their derivation, which is a worthy attribute of a name. Equally suggestive (and avoiding the "code" or "non-code" connotation) would be "typewriter-paired" and "non-typewriter-paired", or "typewriter-compatible" and "typewriter-incompatible".

I point out that "typewriter-paired" and "bit-paired" do have one valuable attribute. Regardless of the connotations that might be drawn in the minds of some regarding their applicability or non-applicability to different classes of keyboard devices, they are well fixed, in the literature, and those who are involved with keyboards, and with keyboard standards know which kind of configuration is meant.

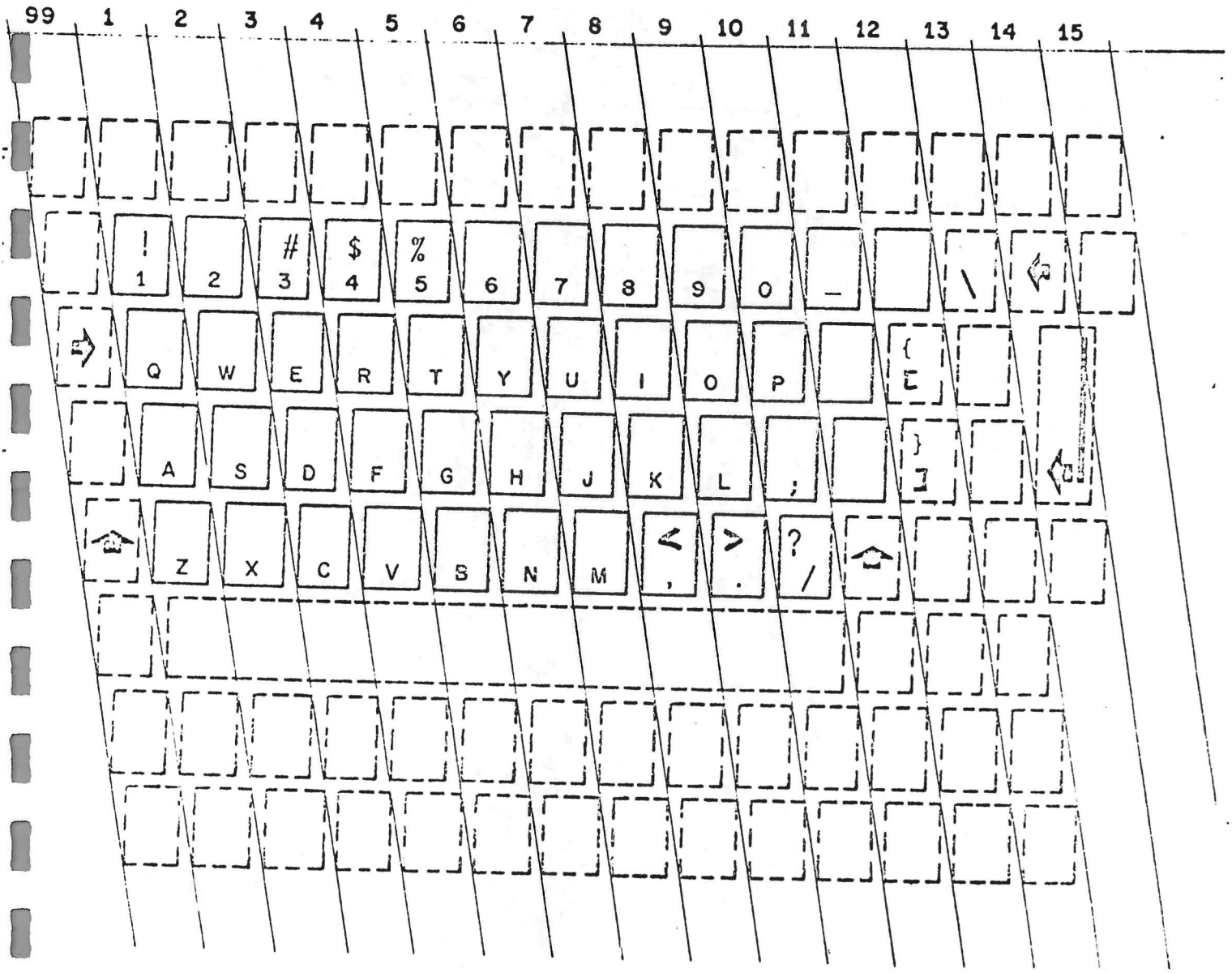
I submit that your concern in this matter might be satisfied equally well, not by a change of name, but by being very specific in the "Scope" section of the proposed standard. For example, to the single sentence which now constitutes the "Scope" might be added.

> "Two basic keyboard arrangements are specified. Each basic keyboard layout is intended for application to keyboard devices, regardless of whether the keyboard device generates or produces coded output, or produces printed copy only".

Yours truly.

C. E. Mackenzie

CEM/jan



E:
PHYSICAL CHARACTERISTICS AND LOCATIONS (i.e., SIZE, SHAPE, SKEW, ETC.) OF THE SPACE BAR OR OF THE KEYS ARE NOT TO BE INFERRED.

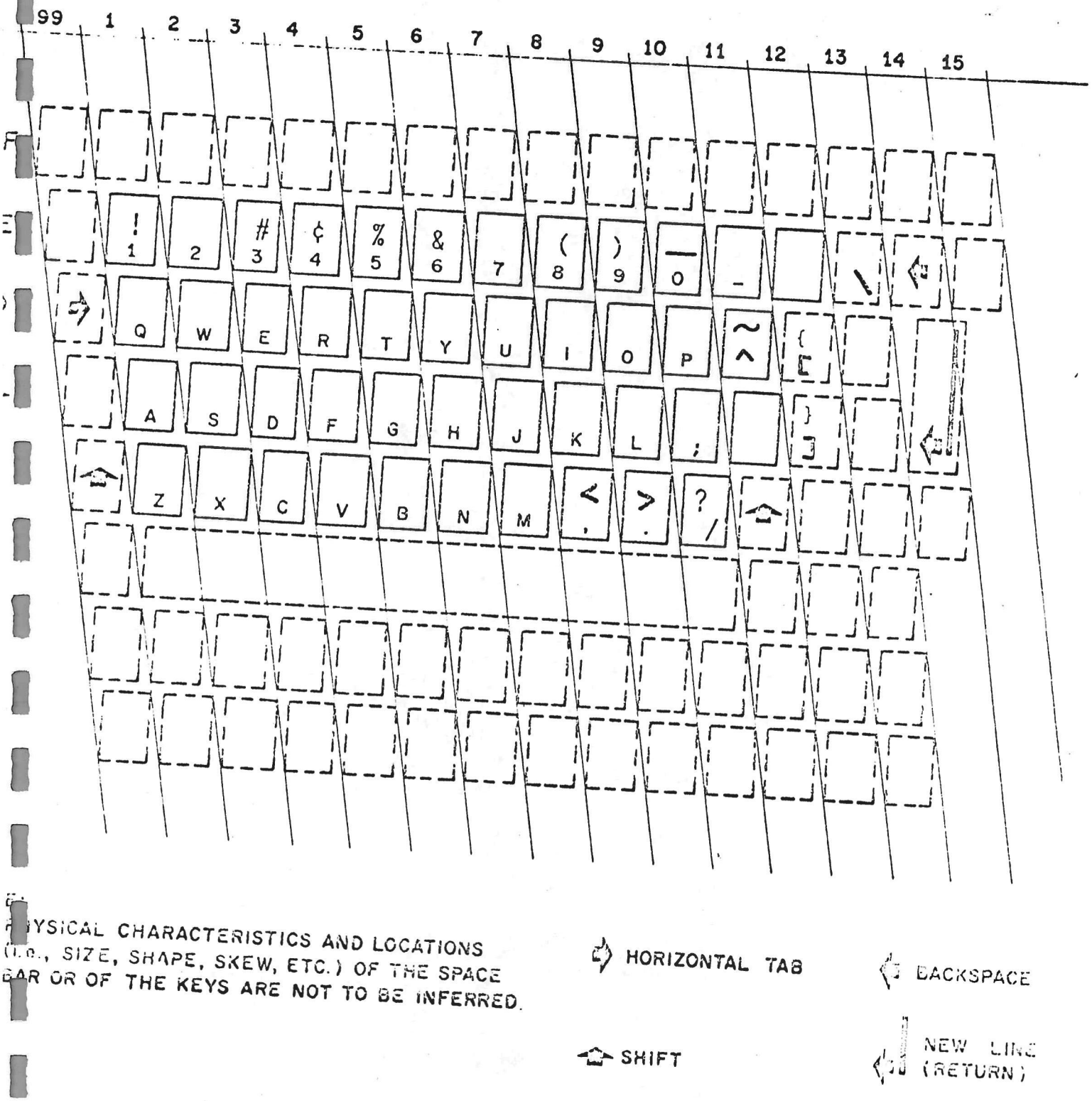
HORIZONTAL TAB

BACKSPACE

SHIFT

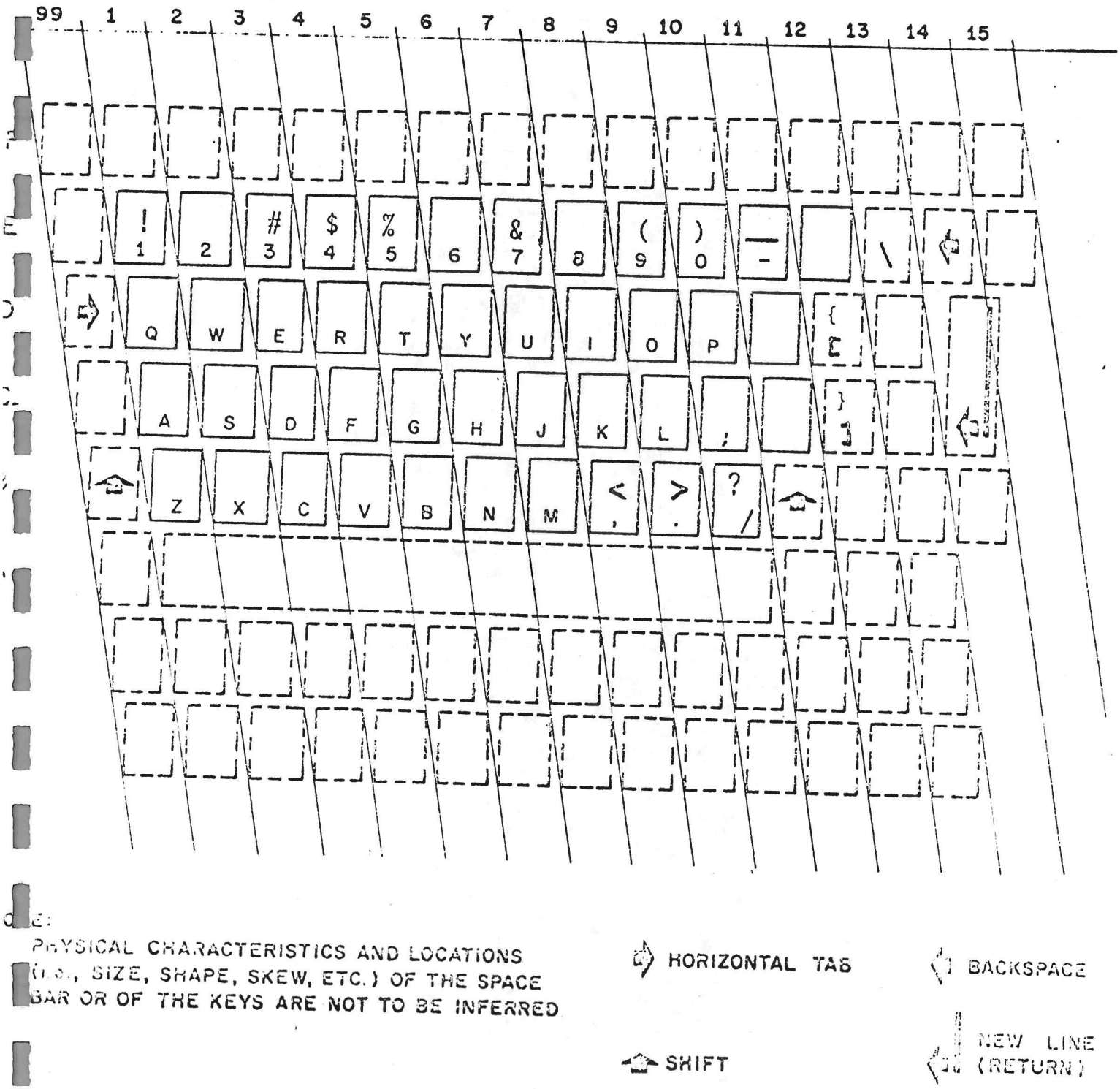
NEW LINE (RETURN)

FIGURE 1: HARD CORE SET



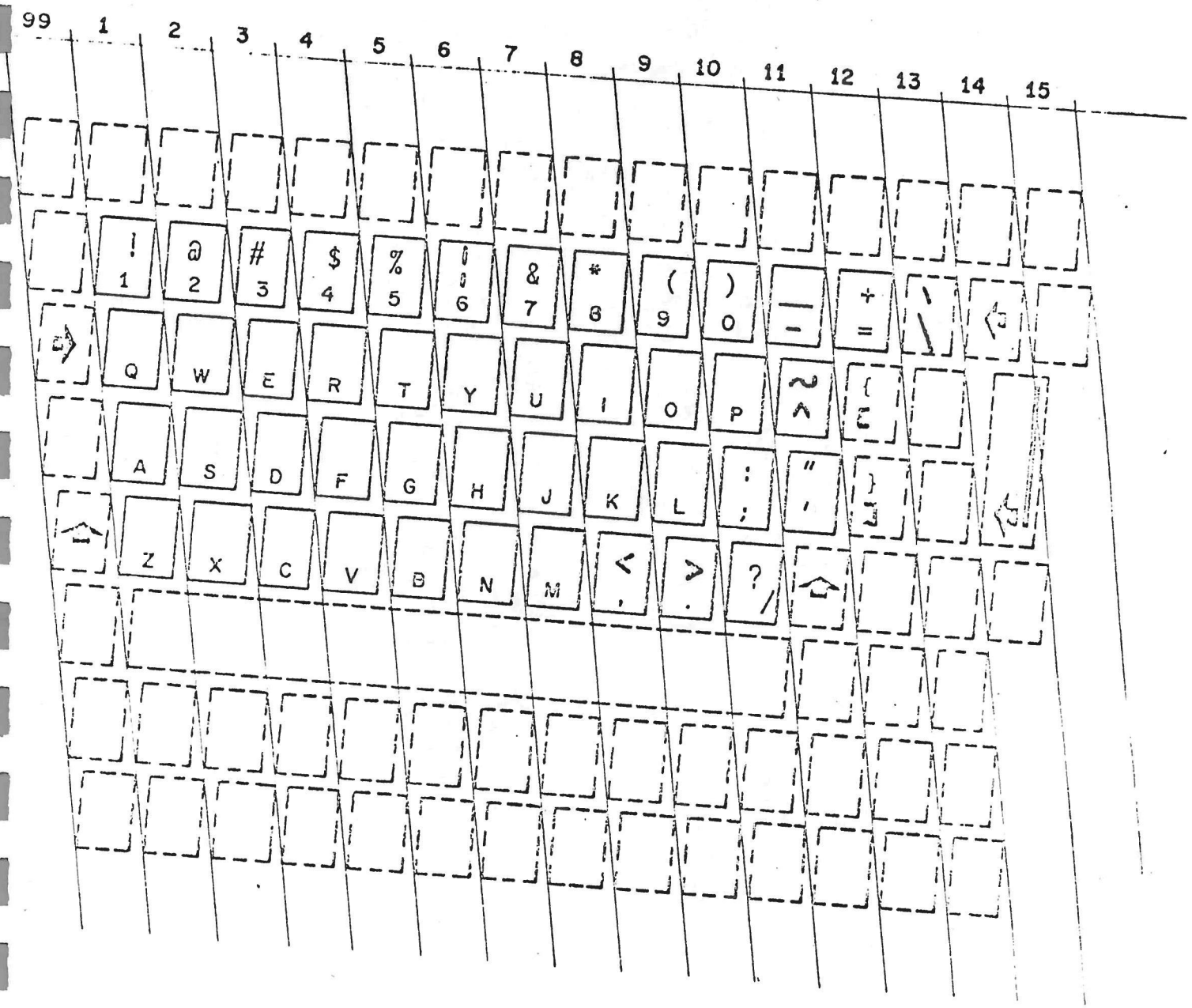
PHYSICAL CHARACTERISTICS AND LOCATIONS (I.E., SIZE, SHAPE, SKEW, ETC.) OF THE SPACE BAR OR OF THE KEYS ARE NOT TO BE INFERRED.

FIGURE 2: BIT-PAIRED, (INCOMPLETE)



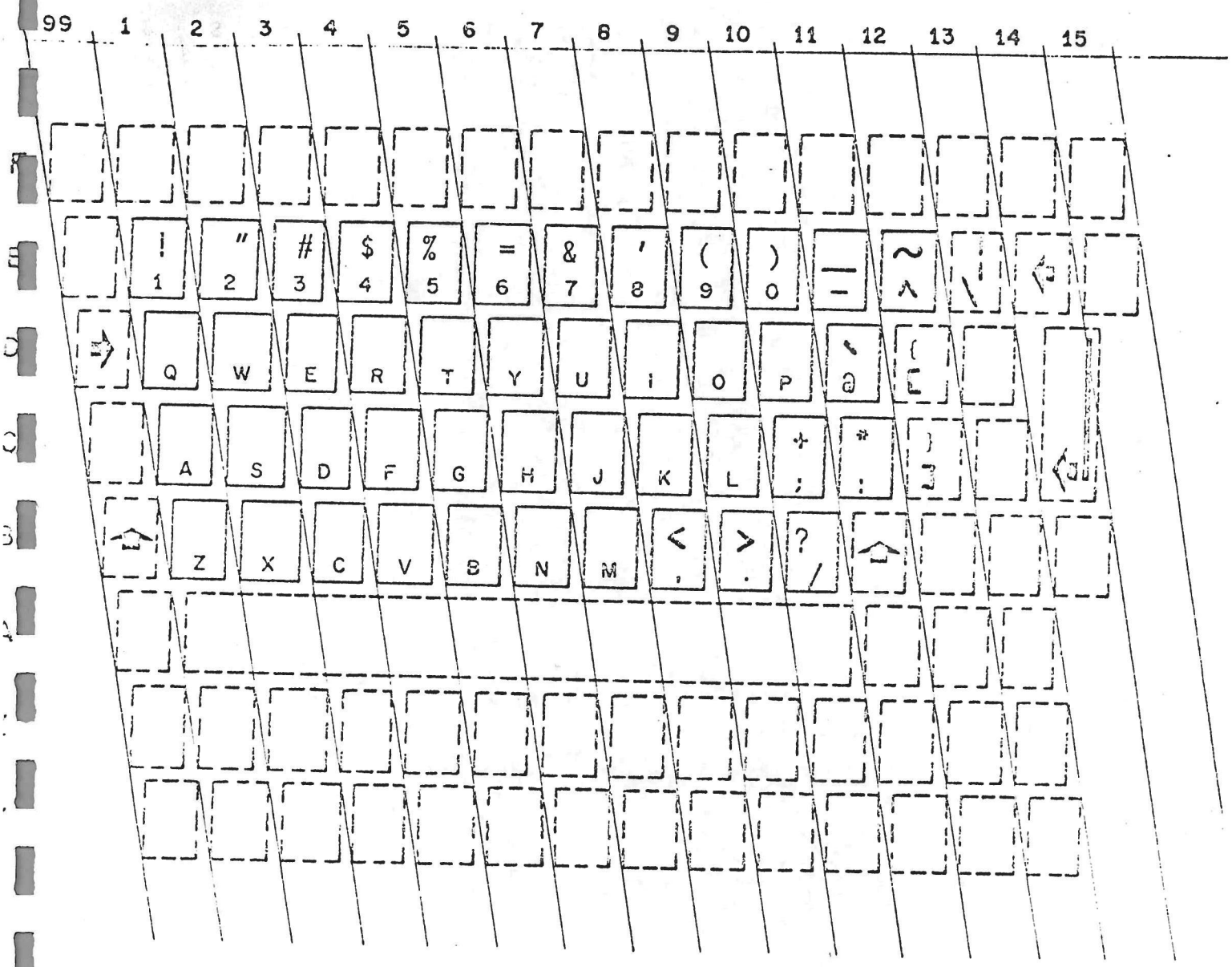
NOTE:
PHYSICAL CHARACTERISTICS AND LOCATIONS (E.G., SIZE, SHAPE, SKEW, ETC.) OF THE SPACE BAR OR OF THE KEYS ARE NOT TO BE INFERRED

FIGURE 3: TYPEWRITER-PAIRED, (INCOMPLETE)



YSICAL CHARACTERISTICS AND LOCATIONS
, SIZE, SHAPE, SKEW, ETC.) OF THE SPACE
AR OR OF THE KEYS ARE NOT TO BE INFERRED

FIGURE 5: TYPEWRITER-PAIRED LAYOUT



NOTE:
PHYSICAL CHARACTERISTICS AND LOCATIONS (i.e., SIZE, SHAPE, SKEW, ETC.) OF THE SPACE BAR OR OF THE KEYS ARE NOT TO BE INFERRED.

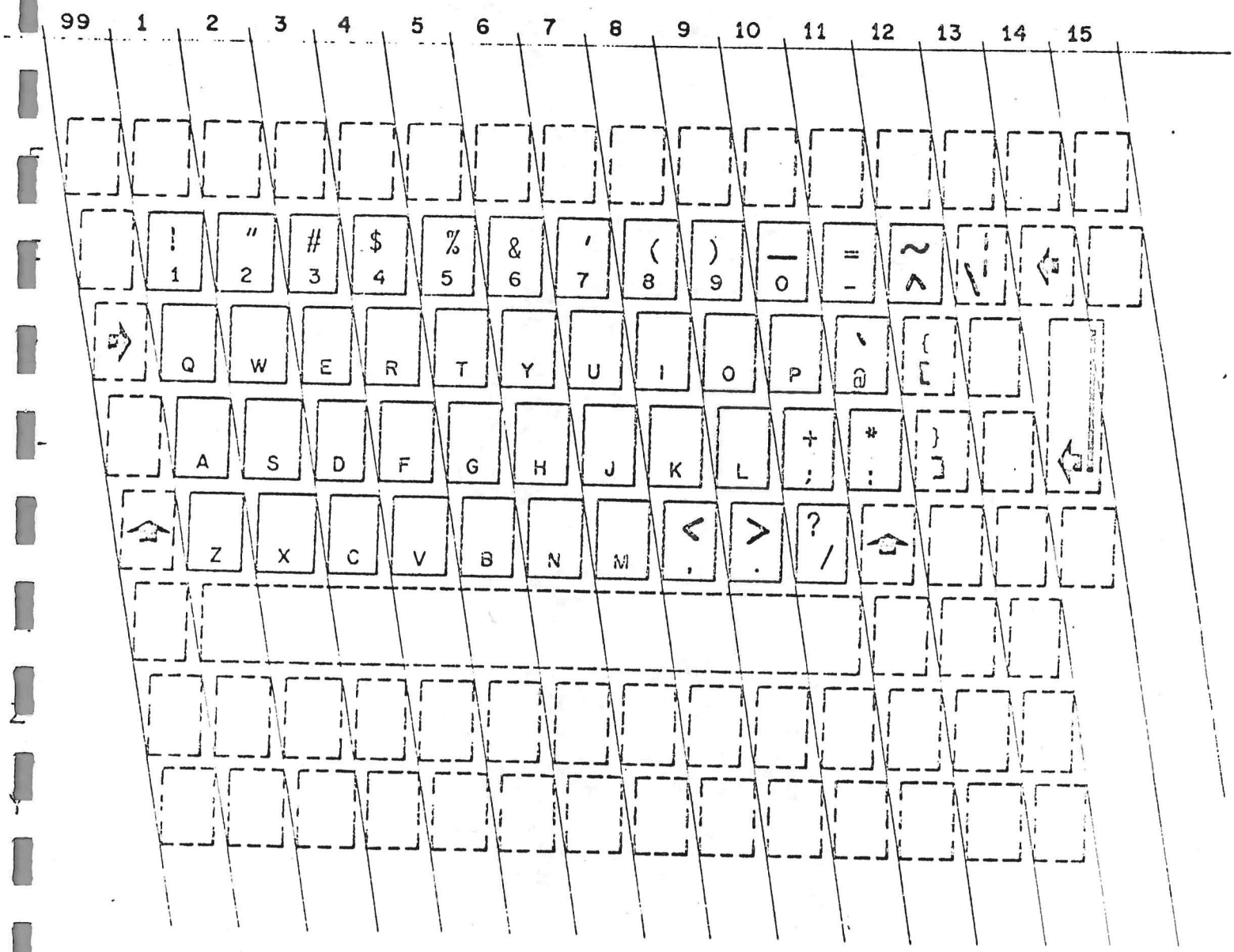
HORIZONTAL TAB

BACKSPACE

SHIFT

NEW LINE (RETURN)

FIGURE 6: COMPROMISE #2



NOTE:
PHYSICAL CHARACTERISTICS AND LOCATIONS (I.E., SIZE, SHAPE, SKEW, ETC.) OF THE SPACE BAR OR OF THE KEYS ARE NOT TO BE INFERRED.

FIGURE 7: BIT-PAIRED LAYOUT